Issue 3 | March 2006

blender art MAGAZINE

Rendering Special !

- Using Yafray in professional environment.
- Learn how to use Sunflow rendering system.
- Yafray caustics explained.
- Learn more about Yafray GI.
- DPI demystified
- Rendering optimization.

www.blenderart.org

blender art MAGAZINE

EDITOR
Gaurav Nawani

MANAGING EDITOR
Sandra Gilbert

WRITERS
Sandra Gilbert
Claas Eicke Kuhnen
Daniel LaBarge
Christopher Kulla
Zsolt Stefan

DESIGN & GRAPHICS
Gaurav Nawani

WEBSITE DESIGN
Nam Pham

www.blenderart.org

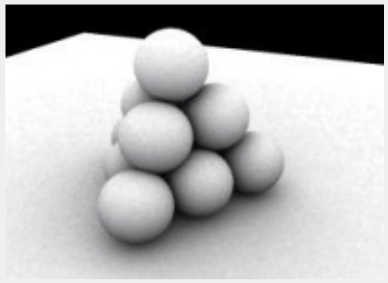
CONTENTS

6

Using Blender for industrial design CAD/CAM

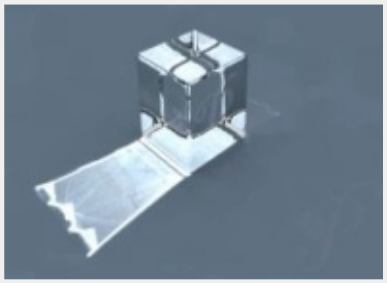
19

GI and lighting in Yafray



13

Using Sunflow rendering system with Blender



25

Using YAfray caustics system from within BLender

Managing editor

Bringing clarity and better understanding to achieve the best render possible.

You've studied all the tutorials, you built the perfect model, added the perfect material, now its time to render your work of art. You hit the big "Render" button, wait for what seems like forever . . . and when it finally finishes . . . well to be honest, the end result looks nothing like what you had imagined. What went wrong?

Most likely, quite a bit went wrong. Rendering is the result of everything that came before and as such there are a lot of factors that can make or break you final image. Is your lighting just right, are your settings correct, did you account for all the special effects you added. Even if you get all the above right, you might spend hours or days waiting for the final render to finish, only to discover you need to tweak something just a little.

Welcome to Issue #3 "Rendering". In this issue we will be covering some of the issues that affect your final render, bringing clarity and better understanding to achieve the best render possible.

In addition to looking into things that can improve your renders, we will also look at ways to decrease your render times as well as take a look at some of the Render engines available for Blender. Of which quite a few have been developed recently and warrant a look see.

We also have 3 separate articles on how to use Yafray, with each showing a different technique and method of use.

Cheers!
Sandra
sandra@blenderart.org

Rendering is an often misunderstood process and until we get the "make render beautiful button" ;) coded, we have to do it the hard way. And that would be by understanding what goes into a render and what affects the rendering process.

The short answer is that EVERYTHING affects the rendering process. That would include materials, lighting, special effects and the scene itself. And while discussing rendering could fill a whole book, for this article we are going to take a look at tips and tricks that can increase or decrease your render times and quality.

When it comes to rendering, some things just take longer. If you just have to have that certain effect, by all means go for it, but expect to wait a while for your render to complete. Here is a list of some things that are liable to increase your render time:
Raytracing, Ambient Occlusion Lighting Setups
Heavy use of particles
Extremely detailed scenes

On the other hand, CG art is full of tricks and work-a rounds to lower render times, such as the use of the fake GI dome. Long used in blender before AO was coded into blender. It is simple to setup, easy to adjust and not as computation heavy as the AO option.

To set one up is easy: Add>Mesh>Icosphere, subdivision 2
Delete the bottom half of the Icosphere
Add a spot light, and parent it to the Icosphere
Hit F7, (Object Settings window) With the Icosphere selected, press the duplivert button.
Select the spot light and adjust the Energy way down (around .01)
Adjust the rest of your settings as needed to get the amount of light and shadow you are looking for.

Other things that can lower render times: Generally use z-buffer lamps over raytraced ones. Set the samples and shadow buffer as low as possible for the quality you looking for.

Similarly, turn the samples down on everything as low as you can get away with that people probably will not notice (lights, subsurf, raydepth, etc.).Render in passes when possible, you can re-composite the passes in GIMP or Photoshop, not only will it lower render times, you will have more control over how each layer goes together with the others.

Only model those details that are needed, UV mapping and proper use of textures can save on your polygon count. When it is all said and done, use only those options and techniques that you actually need to make your image or animation look good. If you can cheat or fake it, then do so, it will save on your render times. If you just have to use it, then be prepared for the wait. While waiting I suggest relaxing with a nice cup of coffee and a good book.

This time Blentuu takes you through the various aspects that affect rendering times. Also a follow up with a few tips on how to optimise your render times.

Blender News

The biggest event in the Blender world is just around the corner. Project Orange is near completion and the premier of “Elephants Dream” has been scheduled for March 24, 2006. Cinema Ketelhuis, a well known cinema for documentaries and art-house movies in Amsterdam, has been chosen as the location for the long awaited premier.

The cinema opens at 17:30 and the showing will start at 18:00. In addition to the screening of “Elephants Dream”, there will also be a preview of the “Making of” documentary. If you miss the first showing, there will be another at 18:45.

After the premier, there is a celebration planned at one of the hippest nightclubs, ironically named “Blender”. While the premier is free of charge, there is an entrance fee to the “after party”. Saturday 25 and Sunday 26 March, the Project Orange team will be on hand all day in Montevideo (Keizersgracht 264, Amsterdam) for further viewing and discussion of the project. For more information on the premier and the following presentations of the movie check http://orange.blender.org/.

Blender has become an ever changing program, with new features constantly been written and revised for better and cleaner functionality. Over the last two months, major coding effort has been focused into the “Noodles” system (node editor) and the render engine. The Noodles system allows for the creation of highly advanced materials and compositing effects. The render engine is being streamlined and modified to work cleaner and more efficiently, allowing for faster and more error free rendering. To check a test build with these new features, drop by the test build forum at www.blender.org.

Since this issue is focusing on rendering, this is a good time to explore all the render engines available for use with blender. In addition to YafRay, POV-Ray, MegaPOV, there are now export scripts available for the following new engines:

Kerkythea
Site:
http://www.softlab.ece.ntua.gr/~jpanta/Graphics/Kerkythea/

Sunflow
Site: http://sunflow.sourceforge.net/

Indigo
Site:
http://homepages.paradise.net.nz/nickamy/indigo.html

3Delight
Site: http://www.3delight.com/

Metropolight
Site: http://www.3dvirtualight.com/mlt/

Using Blender for Industrial Design CAD/CAM

The Making of a Fork

by - Claas Eicke Kuhnen

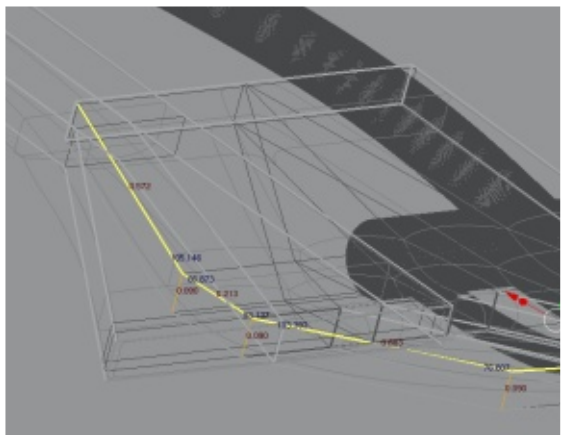
Introduction

A good image is worth a thousand words in Industrial design industry. Blender has become an essential tool for my work, because it allows me to visualize an idea with greater flexibility than any hand sketch, and thus visually communicates my design and concept. While Blender is not made for CAD/CAM and lacks many features those packages frequently include, it is still possible to use it not only for product rendering, but also, to some degree, the whole design process.

Part 1: Modeling

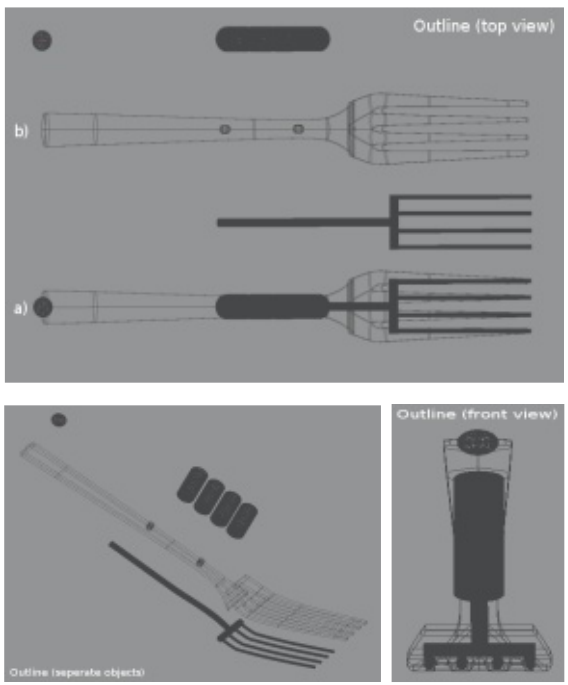
As an object designer with a recent focus on metal smithing, Blender provides me with all the necessary modeling tools to design, explore, and construct my objects. The Polygon tools, together with Subdivided Surfaces, allow me to model all shapes, which I would later be able to construct in metal with my hands. In silver smithing, most shapes are constructed out of sheets, wire, and tubing. Elements can be bent by hand or forged over shapes and then soldered together. This means that I mainly deal with geometric primitive shapes. And those are very easy to model in blender.

The recent addition to calculate the length and angle edges furthers my ability to account for correct measurements and dimensions. This is one of the most important advantages 3D modeling can provide you, because you have the possibility to work in correct proportions and dimensions. Small changes within the design or a change of the point of view can be achieved with a few mouse clicks and do not require the hand sketch to be completely redrawn. (Screenshot: Measurements)



In addition, not only does Blender allow to model mechanical elements, it also makes it possible to animate them, allowing me to proof the concepts and troubleshoot possible problems.

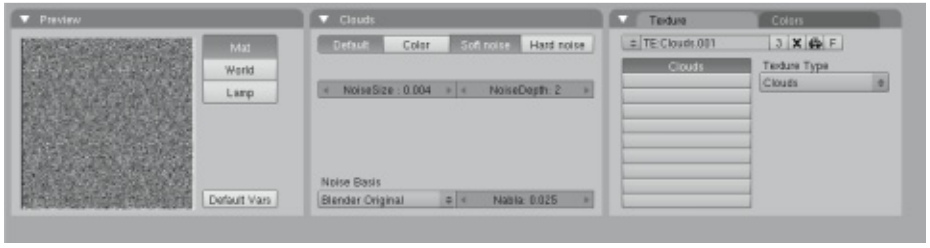
In Orthographic View, screenshots can be taken of the wire frame and composed together. After printing on paper they can be cut out and glued onto the silver sheet, to aid as guides for the shapes I saw in the metal. Working in layers aids a lot in this process because elements can be distributed through them and then selectively shown to create clear and useful screenshots. (Screenshots: Outlines)

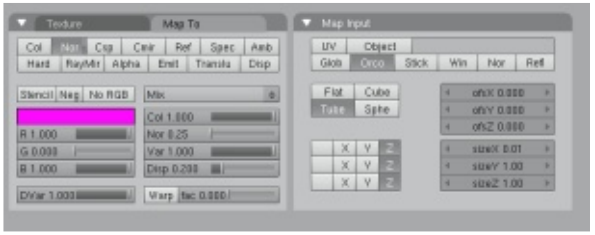


Part 2: Texturing

The current version of Blender and Yafray support most textures and surface light models that an artist needs to create pre-conceived material simulations. However specific materials like brushed metal or translucent materials are still either not yet possible or only possible with additional tricks and time. Specifically in my case, I am unable to acquire those anisotropic reflections and specular highlights, which are required to accurately simulate the fine-brushed surfaces with which I work. But to some degree this feel can be achieved by using a procedural noise map or a painted bump map.

The problem is that you can only reach a certain amount of detail before you will run into moir patterns during renderings. However even a somewhat coarse bump map already provides enough detail to simulate a brushed surface. The texture needs to be stretched along one axis and needs to have the right scale. (Screenshots: Prodecural texture1+Texture2)





Also, a small color tint for the reflections and highlights will reduce the sterile look of raytraced reflections a lot and at the same time give the rendering a feeling which I am used to getting from using real studio lamps. Those lamps do not produce white light. The light is always a bit yellow. It was also important to give the ground plane a different color than plain black. Those hue values were very minimally different from pure black but enough to create a natural look.

Part 3: Rendering

For my renderings, I have the expectation that the final result should be as realistic as possible. I use Yafray as the primary render engine, because it supplies me with HDRi lighting, Global Illumination, Color Bleeding, Caustics, and true Depth of Field. Specifically for my type of work renderings, which mimic the aesthetics of studio shots, these features are essential in communicating what my work is about. HDRi IBL can provide the proper natural light diffusion and colorization, which are sought in object studio photography. In my case, I use an HDRi file which contains only two light areas, simulating the diffused light walls. This will produce extremely soft shadows and reduce the amount of burning highlights.

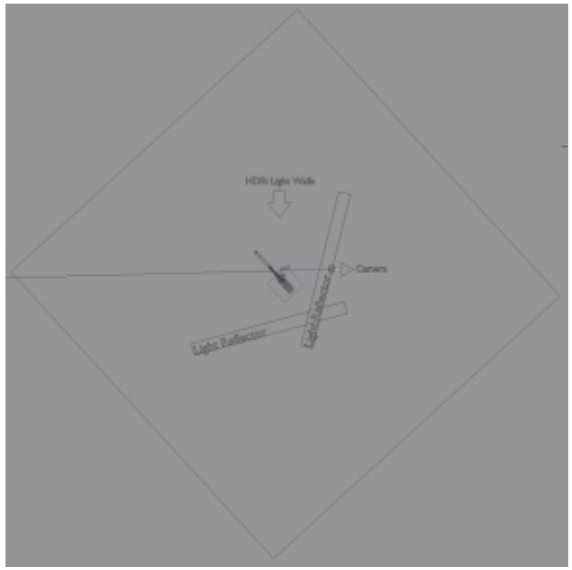
This light probe supplies the desired amount of light from the proper direction in the scene, while eliminating any unwanted reflections (or hot spots) on the work. The HDRi light map (below) will furthermore not only illuminate the front view of the object, but also emit enough low key light to illuminate the metal work from all other directions. This eliminates any need for small light arrays to brighten up specific areas, like the traditional three-key light setup.

Furthermore, the GI option not only accounts for a proper shadow calculation and illumination of the whole geometry, but also increases the realism of the rendering with calculating color bleeding and caustics. Many product renderings lack those fine details needed. Often, it is those fine details that will have a significant impact on the viewer with the realism felt while viewing the renderings.

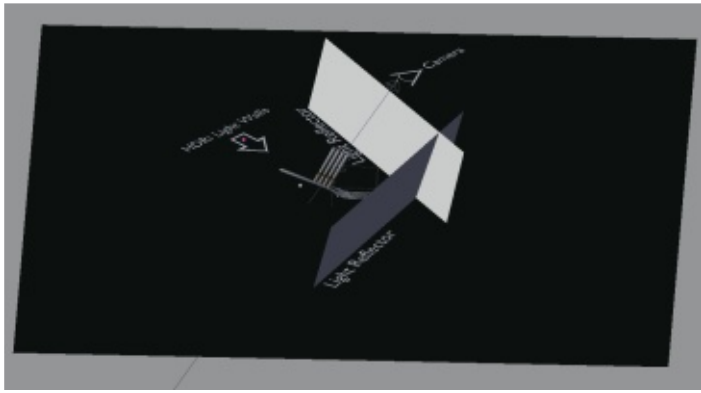
But just selecting an HDRi light map is not always everything you can do. In my situation, I only get light from one side. Because of aesthetic reasons, you want to eliminate or lighten the dark areas so as to increase the overall dynamic range of the rendering. To brighten up shadows and dark areas, white light reflectors are used in real life photography. Same trick applies to digital rendering with GI.

The following two screenshots clearly show the difference. The GI calculates not only the light coming from the HDRi map, but also calculates the light bounced back by the light reflectors and this brightening up of the shadows on the cigarettes also adds more detail to the metal reflections. As you also can see, I use a chrome ball to find out where the light is coming from, that information enable me to position the metal work correctly in the scene.

Rendering: chromeball_test



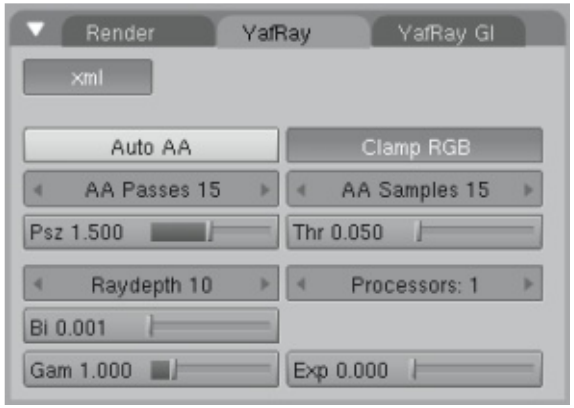
Scene: Topview



Scene: setup3D

Another very important tool for product shots is the Depth of Field. This features is very useful in attracting attention to the specific elements of my work, while other elements are blurred. Yafray supports true DoF, which with higher sample values will produce very smooth results. The render time will obviously increase with the amount of samples, but the end result is more than worth it. The DoF settings can be accessed from within Blender. An aperture value of 0.1 to 0.5 is sufficient. To gain a smooth DoF rendering, in the render settings you need to disable auto AA. Depending on the scene you have to adjust the AA settings, in my scene I used 15 * 15 for AA Passes and AA Samples.

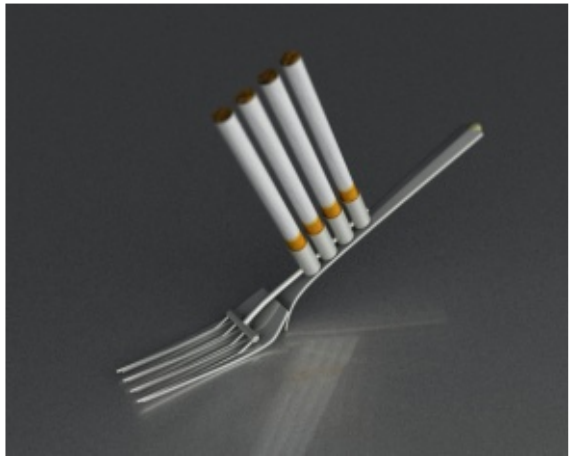
Screenshots next page : AA Settings+DoF_setup



For jewelry renderings, I prefer a textured ground. One of my recent work is shot on a black, semi-reflective surface, which has a fine grain texture. A common mistake made in simulating this effect is to select a too strong reflective value. The reflection on the picture ground should only be slightly visible, so it accompanies the main object and prevents from appearing as a floating object.
Additionally, too sharp reflections will only distract the viewer and steal attention away from the work.

There are two possible ways to achieve this result.

Dirty one, Blender internal:
The reflection can be diffused with a fine noise map in the normal channel. This is a quick way to create blurred reflections, though they are not physically correct. In this case, because the reflection should not get too much attention, this problem is not of any major significance.



Clean one, Yafray
Yafray itself supports the basic features of blurred reflections. Unfortunately, this feature is not accessible from within Blender interface. You need to manually edit the exported XML file and then render from Yafray. Furthermore

Rendering: true_blurred_refl_nobump

Yafray does not shows blurred reflections in other reflections, which might be in some situations a limitation.

To export the scene into an XML file, all you need to do is to enable the XML export, start to render, and stop Yafray the moment it is finished with loading the scene and starting to render(check the terminal window for this during rendering). Open the XML file in any text editor and enter the following shader information into your XML file.

First include the conetrace shader at the beginning of your xml document after the <scene> tag.

```
<shader type="conetrace" name="env"
reflect="on" angle="10"
samples="16" >
   <attributes>
   <color r="0.5" g="0.5" b="0.5" />
   </attributes>
</shader>
```

The sample value will define how fine the blurred reflection is. Higher values above 10 produce the desired smooth result in my scene.

But this also depends on the distance between your camera and the object. In close-ups higher values are needed.

When you use the conetrace shader, you can disable the reflection properties inside blender. In the next step you add "<environment value="env" />" inside the material shader block you want to be reflective. See code box on right.

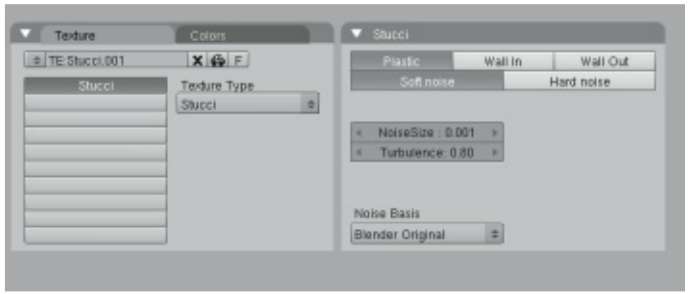
After this step, and saving the file, all you need to do is to let Yafray render the xml file. For my convenience I created a shortcut to Yafray.exe. I only drag and drop the xml file on the shortcut icon and the render process starts automatically.

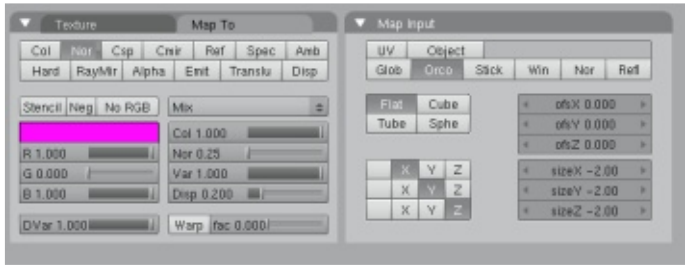
```
<shader type="blendershader"
name="MAMaterial.002" >
               <attributes>
<color r="0.800000" g="0.800000"
b="0.800000" />
               <specular_color
r="1.000000" g="1.000000" b="1.000000"
/>
<mirror_color r="1.000000" g="1.000000"
b="1.000000" />

<diffuse_reflect value="0.800000" />
<specular_amount value="0.500000" />
               <alpha value="1.000000"
/>
               <emit value="0.000000" />
               <matmodes
value="traceable shadow" />
               <diffuse_brdf
value="lambert" />
<specular_brdf
value="blender_cooktorr" />
               <hard value="50" />

<environment value="env" />
               </attributes>
```

But this will result in two problems:
First with using a noise map you will end up with having the noise pattern everywhere on the ground plane. And the noise map does not create the desired surface texture I am looking for. See rendering: fake bump reflection on previous page.

Second, when you only use blurred reflections rendered by Yafray you end up with a surface only showing the reflection but not structure. See rendering: true blurred reflection on previous page.

In my final render I let Yafray render the blurred reflections but I also added a very fine stucci bump map to the normal channel of my ground plane. This will result in first a very nice blurred reflection rendered by Yafray and the stucci bump map will give the ground plane the desired surface structure. This will increase the render time a lot. In case you have a well working bump map for the ground plane, it might not be necessary to use the true blurred reflection within Yafray screenshots of stucci bump material.





You have to play a bit with the reflection values and the material properties because every change for the reflections will also alter the rendered color for the ground plane. At first the reflection and stucci map was too strong and distracting.

In the last rendering I lowerered the reflection value as this also lowered the effect of the stucci bump map.

Conclusion

While Blender lacks many important modeling tools for CAD/CAM, this project clearly shows the strength of Blender as a modeling and design tool and Yafray as a photorealistic render engine. Both are powerful enough to be taken seriously by artists who work in a semi CAD/CAM-like environment similar to the one that I work in and who need a well balanced set of modeling, animation, and rendering tools for the visualization. Because all my shapes are hand crafted, I do not need any tool which is required for rapid prototyping. Also, complex models can be imported from other applications to be textured in Blender and then rendered in Yafray.

In terms of photorealistic output, Yafray already outperforms some of the industry solutions available as long as you use it right. This can be seen as a proof of concept that Open-Source software can realistically compete with commercial software solutions. While there are of course render and material features Blender and Yafray cannot support at the moment, it is important to keep in mind that both programs are currently undergoing a serious reconstruction.

Blender is getting a new node-based material system and Yafray continues to improve the physical accuracy in rendering. As an artist, I hope that features already present in Yafray (such as blurred reflections) will be further integrated and more accessible from Blender. Those material and render options not only increase the potential of this software combination, but continue to close the gap between Open-Source and commercial software. ■

The real fork model (photograph)

Claas Eicke Kuhnen (aka F.ip2)

I use blender for my general artwork, but mainly to visualize my silverwork to explore the design concept and proof interactions. I always try to achieve photorealism in my work and explore new technology and approaches.

I worked during my undergraduate as a graphic and media designer. Together with my brother we founded ConColori and provide industrial design solutions.

www.concolori.de
Fip2@gmx.net
ClaasKuhnen@concolori.de

Using the Sunflow rendering system with Blender

by Christopher Kulla

Introduction

In this tutorial, I will present some of the basic features of Sunflow and how to use it from Blender.

Sunflow is my open source rendering system. It is built around a small but very flexible core which allows it to be completely customized. It contains several modern algorithms that make it a good candidate for generating realistic images. It has been recently connected to Blender via a simple export script which I will be describing.

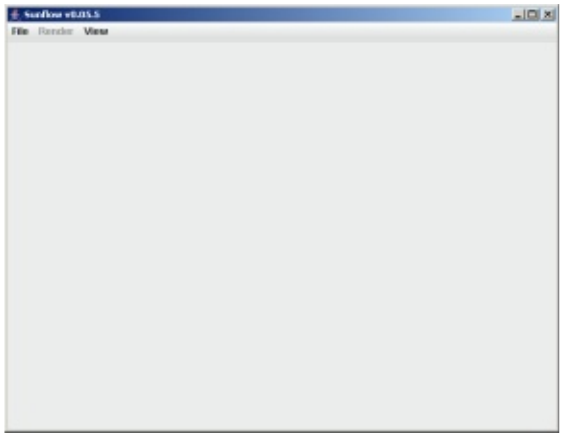
Installation

Sunflow is written in Java. This means you will need to download and install the Sun JRE (or JDK) if you don't have it already. Go to http://java.sun.com/ and follow the instructions for your particular operating system (OS X users will need to go to the Apple website). Sunflow requires Java version 5 (sometimes called 1.5) or higher.

Once you are setup, go to a command prompt and type: java -version

If the correct version number appears everything is setup correctly.

Next, get the binary release from the Sunflow website (v0.05.5 at the time of this writing) and unzip it to a directory of your choice. Go to that directory in command line mode and type the command line described in the README file included in the distribution. You should see the following window appear:

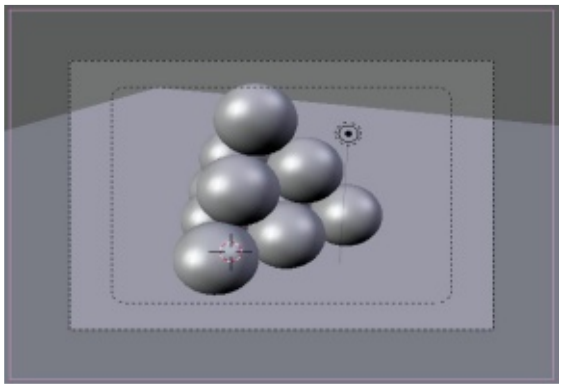


Finally, get the blender exporter script from the Sunflow website and install it into to your blenders (Inside your blender installation folder) scripts folder. Next when you start Blender you should see an option appear in the File->Export menu as follows:

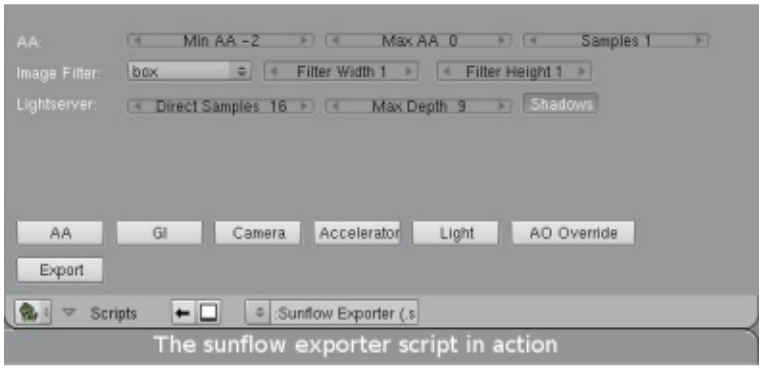
You are now ready to render your first image!

First Render: Ambient Occlusion

I have prepared a very simple scene here for illustration purposes. Feel free to follow along with your own favorite model. Here is what my scene looks like in the 3D view:



Go to the File -> Export and pick the Sunflow Exporter you just installed. It will bring up the following UI in the script window:



The sunflow exporter script in action

We are just going to change a few settings the first time around. Our goal is to do a simple ambient occlusion render to get a feel for the process. Don't worry about the many different settings you see, we will cover most of them soon.

In the AO Override tab, click the 'Amb Occ' toggle. This is simply telling the exporter that we want to override all shading in the scene with the prescribed ambient occlusion shader.

The most important options are samples (number of rays traced to calculate the occlusion) and the distance setting (how far to look for other objects).
For now, leave the number of samples as it is, and set the distance to a reasonable value for your scene. The default of zero corresponds to infinitely long rays.

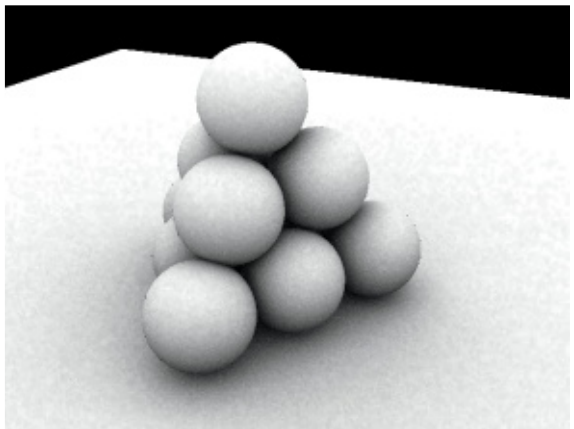
Now hit the Export button and pick a location to put the exported scene file. You can look at Blender's output window to see the steps the exporter is taking. It may take a few seconds to walk through your scene if you have many objects. When the file is exported you will be sent back to the exporter options.

Now go to the Sunflow window you launched earlier and open the scene file you just wrote out. If loading succeeds without errors, the Render menu option will now be enabled.

Select Render -> IPR first. The image will start rendering in a very blocky form and progressively refine itself. After just a few seconds, you should already see what the final image is going to be like. You may cancel the render at anytime. In most cases, less than 10% of the image is needed before you are ready to launch the full resolution render.

Now select Render -> Render. The image will render in small buckets this time. If you have a multi-processor machine you will see several buckets rendering at once. After a few seconds you should see something like the scene below.

Congratulations! You just rendered your first image with Sunflow.



Your image will probably be noisy and have many small artifacts. Don't worry. We will learn how to enhance the quality of the image soon.

Now, let's make our scene a bit more interesting by adding some lights and materials to it!

Lights and Materials

Go back to blender and assign some unique materials to each surface. Keep in mind that the exporter does not yet support the full range of Blender shaders and textures.

For now, we will just be creating regular diffuse materials. For each material, simply assign a unique color. This is the only attribute that is read by default, all other options are ignored. There are a few exceptions to this rule to generate other Sunflow shaders, but we will cover them later on.

Next, we need some lights. The exporter currently only supports lamps and square area lights. Area lights will give you nice soft shadows at the expense of additional render time. My scene with basic shaders and lights now looks like on the image at right.

Once you are happy with your scene, time to export again! This time, let's try to understand the various settings and how they affect image quality.

The most important are probably the AA settings. The min and max values control how much computation goes into each pixel. Negative values mean computations will be interpolated across several pixels, positive values mean more calculations will be done per pixel. Zero means a single computation is done per pixel. For example: the default values of -2 and 0 mean that the renderer will start by calculating every 4th pixel, and adaptively calculate down to the single pixel level where it is needed. This time, try setting your AA values to 0 and 2. Each pixel will now be shaded at least once, and up to 16 times (level 2).

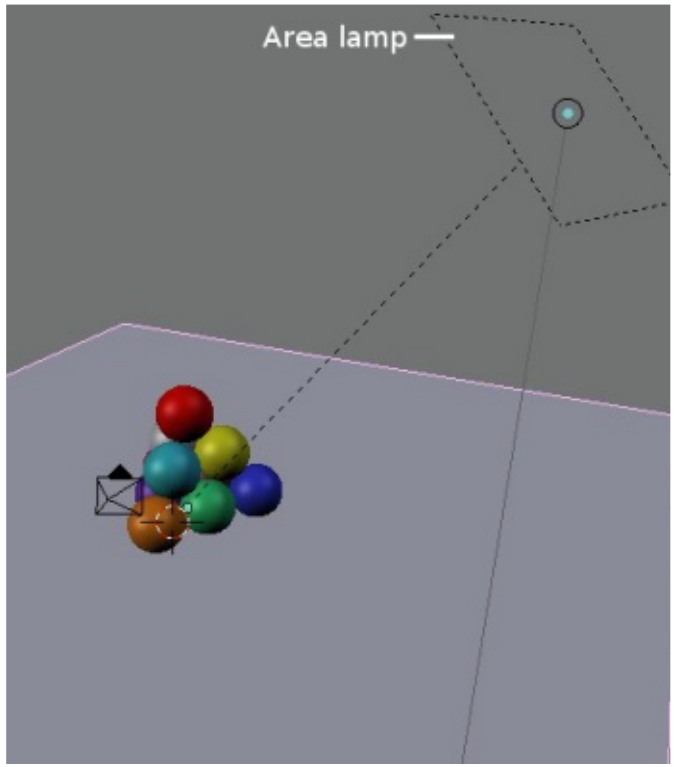
If we calculate more than one value per pixel, we will need to average them together. The way this is done is controlled by the image filter setting. Let's pick the 'Mitchell' filter as it provides a nice sharpening of the image. This particular filter has a fixed width and height, so those settings will disappear from the dialog when it is selected.

Light server settings let you control how many rays are used per light. The default of 16 is a bit high but usually gives good quality.

The GI tab controls global illumination. This is an advanced topic we will explore in the next section. Leave all the controls at their default values for now.

The Accelerator tab contains settings which have no impact on image quality but can make drastic differences in image calculation time. The default accelerator is the kdtree and is usually the best choice.

You can experiment with different accelerator types and see which one provides the fastest results for your scene. You can also adjust the bucket order in this tab. Switch to spiral this time to have the image computed from the center out.



Under the light tab, we have the option of adjusting the strength of all lights in the scene. Since Sunflow uses more physically based calculations than Blender, the light powers will not usually match. This global scale factor can help you compensate for this. I set mine to 5, but this will vary from scene to scene. Note that the scale factor only affects area lights.

Finally, remember to deselect the 'Amb Occ' button from the AO Override tab if it was still selected from the previous export.

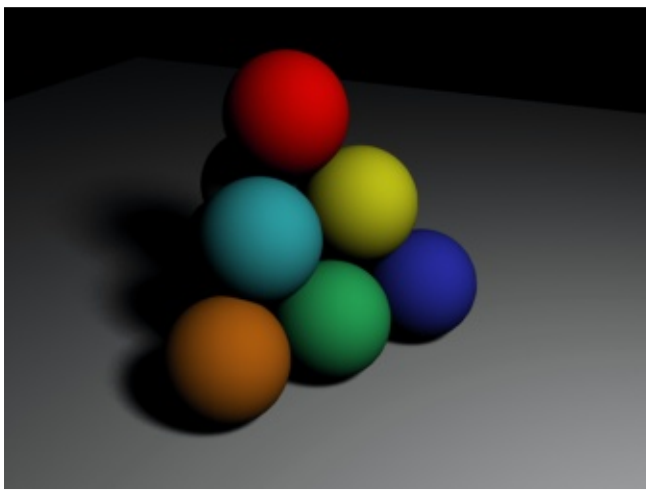
We can now export and render again. If you export to the same location as before (overwriting the old file), you can simply use the File -> Re-open menu option in the Sunflow GUI to quickly reload your scene.

Use the IPR feature again to see what your scene looks like quickly. You will probably need a couple of export cycles before you get the light intensities just right. Here is my result (Image left bottom).

Despite the soft shadows, this image isn't very realistic yet. I will show you how to address this in the next section.

Global Illumination Techniques

As I mentioned in the introduction, Sunflow is a very customizable renderer. It is therefore able to incorporate many different GI algorithms into one coherent framework. This gives the artists different ways to achieve their goals and makes it possible for developers to introduce new algorithms, without breaking older scenes.



I will now briefly describe one of the methods of calculating global illumination: path tracing. This is the most accurate and easiest method to learn. The catch is that it can take quite long to achieve high quality results.
In the exporter, go to the GI tab and enable the path button. The algorithm is controlled by two parameters: samples and bounces. The first default to 32 and controls the quality of the result. The second default to 1 and controls how many bounces the light is allowed to make.

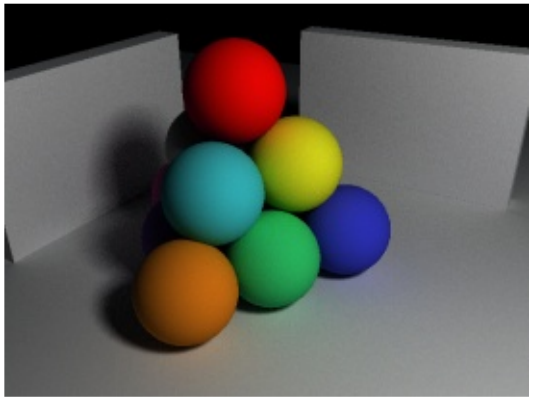
Leave the values at their defaults for now to see their effect. Since this algorithm will produce very noisy results, go to the AA tab and set both the min and max to 0. This will force only one sample to be taken per pixel, which avoids the anti-aliasing from being forced to super sample every single pixel because of the noise.

You may also want to set the filter back to box (width and height 1) to avoid introducing artifacts by filtering the noise.

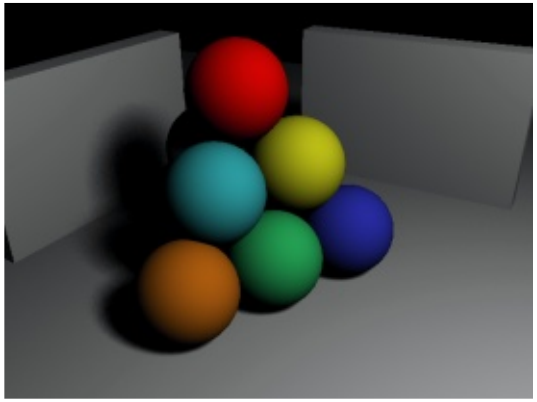
Export and render again. I added some walls behind my scene to see the effect of the bounces. Compare these two renders with and without global illumination:

What is Global Illumination

Global illumination refers to methods which are based on physical computation of lighting in a scene by taking into account the effects of light bounces on and off the 3d surfaces. It provides the most realistic lighting simulation of in CG.



Scene with GI



Scene with no GI

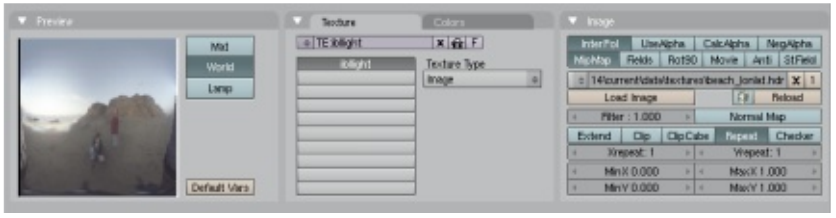
Notice how the shadowed regions are no longer completely dark and how we obtained some subtle color bleeding on the floor

and between the objects.
You may find it hard to get rid of the noise in the image in a reasonable amount of time, even if you make the number of samples quite high. This is a theoretical limitation of the path tracing algorithm. Photon maps and irradiance caching are two algorithms that when combined can yield a much more efficient solution; however they are much trickier to setup, even for experienced users.

Rather than go into the details of these advanced methods, it might be better to know that simpler and more efficient algorithms exist that will very likely appear in Sunflow soon.

Image based lighting

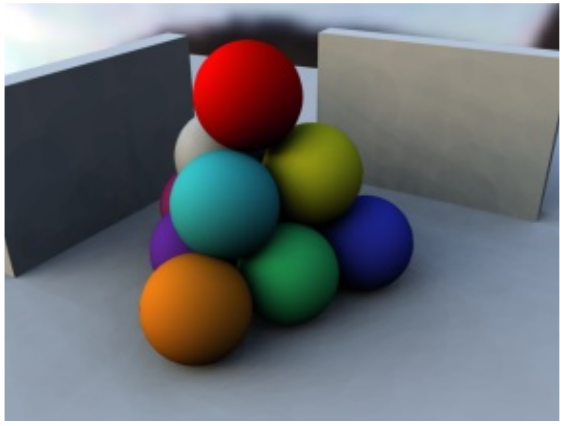
In this section we will cover the use of HDRI environment maps for lighting. Back in your blender scene, load in a longitude-latitude HDRI environment map as a texture at the world level. Be sure to uncheck 'relative paths' when loading the map. Rename the texture object to 'iblight' as shown. This lets the exporter know you want to treat this map as an environment light source.



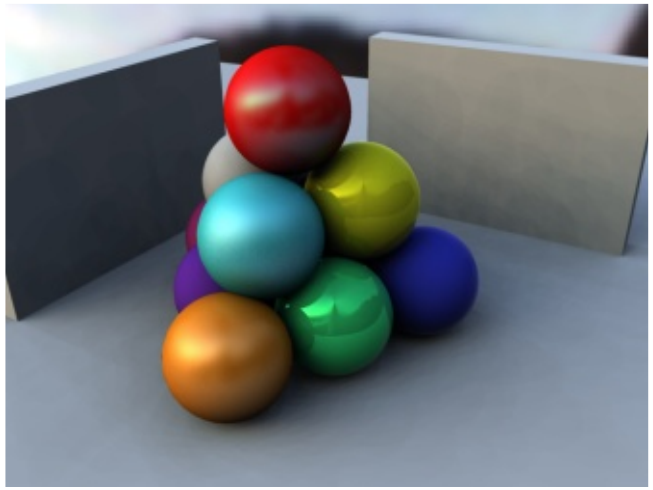
Go ahead and hide all the lights you were using before (the exporter respects Blender layers), and make sure your scene is 'open'. This type of lighting assumes the light is coming from infinitely far away, so its best suited to outdoor renderings. Indoor scenes will perform very poorly with this technique.

Sunflow approximates the environment map with a fixed number of directional lights. This is controlled by the 'Direct samples' setting in the exporter. Using the default of 16 you will probably see some sharp edges to the shadows from the individual lights used. Simply increase the number of light samples until the sharp shadow boundaries are decreased to an acceptable level.

Exporting and rendering again, you should end up with something like this:



This image is quite realistic, even though we have not used any global illumination. Of course you may enable path tracing as we did before to simulate the extra bounces of light.



Glossy materials

Finally, let's make our shaders a little more interesting. Up until now we have used completely diffuse shaders.

Let's start with the shiny shader. It simply adds a thin reflection coat on top of the diffuse color. Rename one of your objects materials so that its name starts with 'sfshiny'. The exporter will now read the 'RayMirr' attribute to determine the amount of reflection to apply.

Next is the phong shader which provides blurry reflections. Rename the material you wish to edit so that it starts with 'sfphong'. The specular color now controls the amount of reflection while the 'hard' value controls the blurriness. Higher hardness values result in sharper reflections. For my scene I used values ranging from 5 to 50.

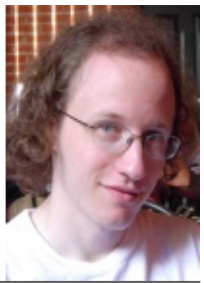
Now that we have added several layers of reflections, we need to pay attention to the 'Max Depth' setting in the exporter dialog. This controls how many levels of reflections are allowed. The default of 9 is rather high so we will reduce it to 3 this time.

Here is the final result, rendered with the image based lighting from the previous section:

Conclusion

I hope you've enjoyed this quick tour of some the features of the Sunflow rendering system.

One tutorial is not enough to cover everything that is possible with this software. Many more things are supported, like textures, refractions, depth of field, caustics, and even procedural shading. Please visit the Sunflow homepage (http://sunflow.sourceforge.net) and the Elysiun forums (http://www.elysiun.com/) to learn more. Happy rendering! ■



Christopher Kulla

Christopher Kulla is the primary developer of the Sunflow Rendering system. In his day job he works as a software developer for Reel FX Creative Studios in Dallas, Texas, working on commercials, direct-to-DVD and animated feature film projects.

http://sunflow.sourceforge.net

GI and HDRI lighting in Yafray

by Zsolt Stephan

Introduction

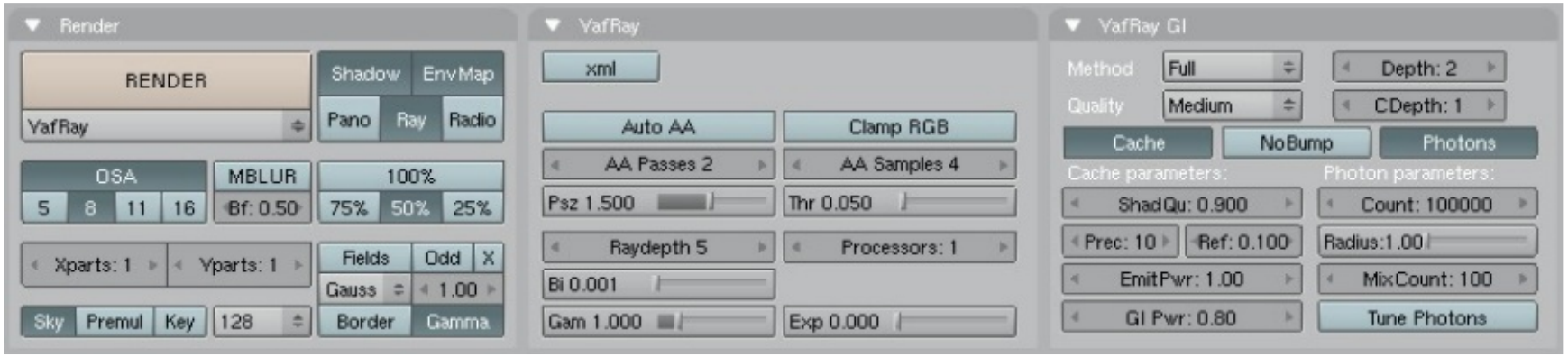
There are a multitude of ways to light a scene, from a few simple spotlights, through arealights and complex light rigs upto different forms of global illumination. Here I will be discussing this latter option using Yafray for rendering, from inside Blender. First I'd like to address the common misconception that GI is some magic 'make realistic rendering' button. It is simply a type of lighting model that is more advanced than the good old spot lamps, and more physically accurate. It will not automatically make a nice image for you, it will simply give you a head start by being a substitution for a very complex lighting rig with dozens of lamps that would need to be placed by hand. It therefore saves time (and time=money). It basically gives you a base fill light, with an even light distribution, and a nice natural look. This can (and should) be complemented with traditional lights.

Besides the lighting, the GI rendering algorithm is also responsible for colour bleeding between objects (hard to fake without GI), and caustics (near impossible to fake). However, GI is slower than raytracing and much slower than scanline rendering, and is therefore only for final rendering for stills. It can't really be used for animations (yet), not only because of rendering time, but because light/dark spots are randomly rendered (sampled) and they jump around from frame to frame.

First we'll look at the GI settings for Yafray, and then I'll describe a typical lighting setup that I use for product visualization.

Global illumination is different from raytracing. In the latter a ray is traced from the camera to an object, this can be reflected/refracted, and then traced to all lights that are visible. This gives you reflection, refraction and (hard) shadows. GI, on the other hand traces the path of the photons themselves from the lamps to the objects, bouncing between objects in the scene. This gives you reflection, refraction AND colour bleeding, soft shadows, caustics (impossible with raytracing) and generally a nicer look. The origin of the light in GI can be a lamp, or any object. For outside scenes, the whole backdrop (world) can emit light. Since Blender does not support GI yet, I use Yafray.

The extended Yafray panels in the Blender rendering [F10] options.



GI settings

First, make sure that you have Yafray installed, and its location added to your OS's 'path' variable (see the Yafray readme!). In Blender, go to the render buttons, and in the drop-down list under the large RENDER button, select Yafray. This should open up two extra settings panels: "Yafray" and "Yafray GI". In the Yafray panel, the XML button should be turned off, this allows you to see the render in progress.

The two buttons below allow you to specify the antialiasing. Other buttons below these aren't important now, they don't need to be changed. The other panel, "Yafray GI" is more important. The "Method" drop-down list allows you to select either None for no GI, SkyDome for a method similar to Blender's Ambient Occlusion, and Full for the proper global illumination rendering. Selecting Full will open up new buttons on the bottom.

The Quality setting will determine the (guess what!) quality as well as speed of the render, a setting of medium/high should be enough for most situations. If there are small faces, detailed models, or surfaces very close to each other, you might need to set the quality higher to get rid of render artifacts.

A complete GI solution can be painfully slow to render (hours), so some tricks can be used to speed it up. Turn on both the Cache and the Photons buttons in the Yafray GI panel. The photon cache will speed up the render over ten times! However, it will lead to some bright/dark spots on the final render. Increase the Quality to get rid of these, or change the "Ref" button. If you set the "Ref" (refinement) to anything other than 1 (I use 0.1), it will calculate the photon map two times before the final render. The second pass does more calculations in the parts of the picture that have more detail, and the final result will therefore be better.

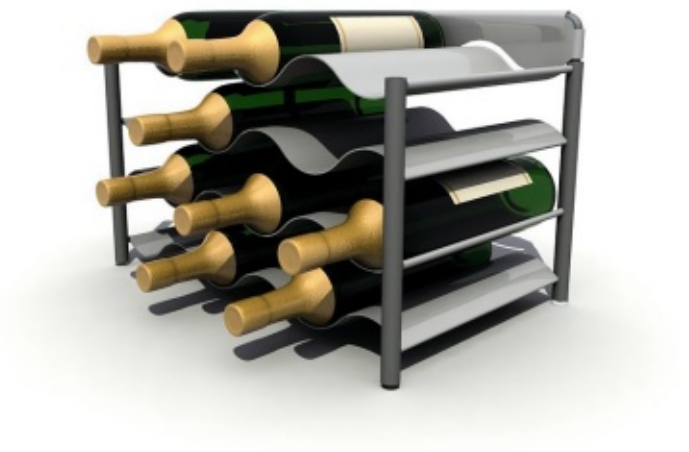
The other important settings are: "EmitPwr" and "GI Pwr". The emitpower determines how bright emitting objects, arealights and the background are. GI power determines how much "energy" (light) these emitters will emit. It is usually enough to change the GI Power setting to make your scene darker or brighter.

Emitters

To use GI, you need to have some light emitters in your scene. Light emitters in Blender/Yafray can be the following: arealights, objects with a material Emit value more than 0, and the background. The simplest thing to do is set the background to any colour that isn't black. The background will emit from all directions evenly, similar to an overcast day outside. You can also load a HDR light probe, which determines what coloured light comes from which direction.

Now, lets try some lighting! I will use a scene I did not too long ago, it is a design for a fold-up wine holder. You can see the final image here, it is for product visualization. Therefore the most important part of the image is the product itself, and nothing else should be present to draw the viewers' attention away, not even a background scene.

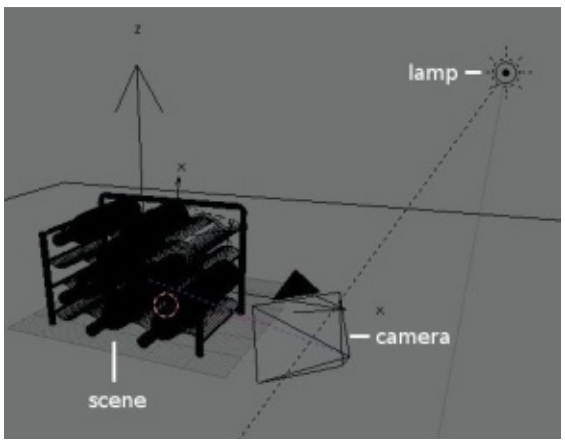
Product Visualization I - "White Space"



This is one of my favorite setups, for lack of a better term I called it "white space", the product is on a completely white ground, with a white background, yet it doesn't just float in space, it has shadows. It looks very good printed on paper, as well as in any document or other white background (a webpage for example), creating the impression that the object itself is sitting there on the paper.

If you look closer, you can see there are two types of shadow here. The hard shadow comes from a sun light, but you can use any other type of light that casts shadows. NOTE: except arealights! Arealights act as photon emitters when GI rendering in Yafray, and not as normal lights! The nice soft shadows are the result of GI.

The scene



The scene, besides the product itself, is simply a white plane, and a sun lamp, nothing more, very simple.

The ground.

The white plane doesn't extend too far, just far enough to catch all the shadows of the objects. It is completely white, and the Reflectivity is 1.0. Note that the option "Only shadow" doesn't work with Yafray. This means the edges of the plane would be visible in the final render, which is not good! This can be fixed in the render settings, see below.

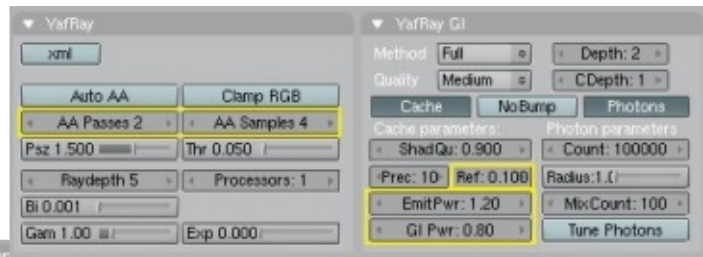
The light

There is only one light in the scene, a sun lamp with Raytraced shadows turned on. It is slightly yellowish in colour, with a power of 0.6.

The World

Very simple again, the world is one coloured, a very slightly blueish white.

Render settings



IMPORTANT! It is really lame, and I have seen bad examples of it many times, when a product is showcased on a grey plane, floating in space, or at least on some sort of background. This grey plane shouts CGI. It's important that you can't see this base plane, it should blend into the background. In this example I used a white plane, but even a white plane sometimes is rendered grey at the edges. The background itself is slightly blue and not white. So the trick to have this "White Space" effect is to set the EmitPwr in the Yafray GI settings above 1.

It takes some experimentation for the right settings, so that you can't see the edge of the base plane, but it doesn't glow in an unnatural way either. A setting of 1.20 seems to work well. GI power is 0.80. If you make the sun lamp brighter, then decrease these settings, if its darker then increase these.

Make some low resolution test renders. Click and hold the left mouse button on the render, this will show the colour of the given pixel in the bottom left corner. Drag the mouse around, and make sure that both the background, and the base plane is R: 255, G: 255, B: 255. Then you're ready for the high res final render! Make sure to set the quality to Medium or High, the Ref value to something other than 1, and turn on Antialiasing, in this example I used 2 passes with 4 samples each, this is the setting I usually use, it is a good quality/speed trade off. And see how much quality we have achieved with a very simple setup!

Product Visualization II - HDR Lighting
Instead of a plain white-ish world colour, you can set a High Dynamic Range light probe for the background. This will vary the light colour

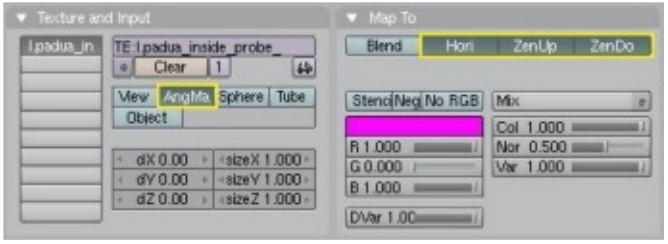
and intensity across the virtual sky, making a more natural, smooth look. To clarify the two terms, HDR means high-dynamic range image, where instead of RGB values from 0 to 255, floating point numbers with better precision, and a higher "dynamic range" are used to store the luminosity (light emission) of each pixel. Light probe means the image is a complete 360-degree panoramic picture of the environment. The process of making HDR light probes is outside the scope of this tutorial, so I will just say either download one of the several HDR probes available on the net (see http://www.debevec.org for some starters), or make your own. Make sure it is in the light probe format (aka. angular map), as opposed to: vertical cross, latitude/longitude, or some other panoramic format.

This next step can be skipped, but I highly recommend it. The problem with GI illumination with Yafray is that for a a fast rendering, it uses some random samples of the environment. A nice and sharp HDR image can have differing light values in a short interval, ie: too much contrast.

With just a few samples, some samples will fall onto a very bright pixel, and some will fall onto a nearby darker pixel, and this will result in a "spotty" render. I solve this problem by blurring the original HDR map. From http://www.hdrshop.org, you can download HDRShop v1 for free. Open your light probe, and in the Filters menu, select Gaussian blur, and blur the image, using a value of 4-5 for smaller maps: 512*512, to 12-15 for higher resolution maps, like 1512*1512. Resave it with a different name.

Loading the HDR probe

Go to the texture buttons, and select the green "World" button. Set texture type to Image, and load the *.hdr file like any other image. Go to the world button, and in the Map To panel, set mapping to Angular map, and the output to Hori, ZenUp and ZenDown. Yafray will automatically see that you have a HDR world texture, and when GI is turned on, it automatically uses that image for lighting.

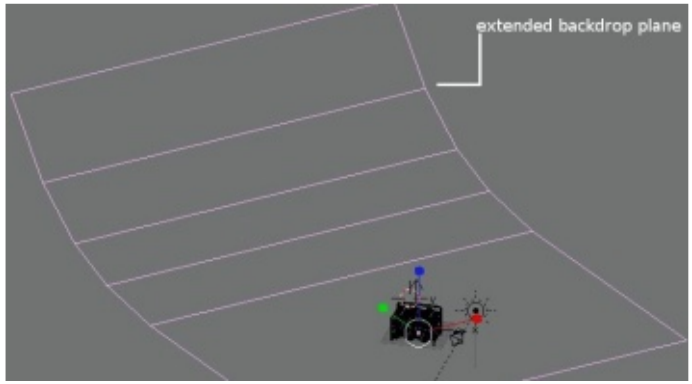


Scene setup

Similar to the previous example. The base plane is not enough however, as you will see its edge in the render. To fix this, make the farther end of the plane curve up into the air, high enough to block all of the camera's filed of view. Make sure this curve starts farther than the shadows of the object, otherwise you will see the shadow curve up unnaturally into the air. See top right image.

Lights

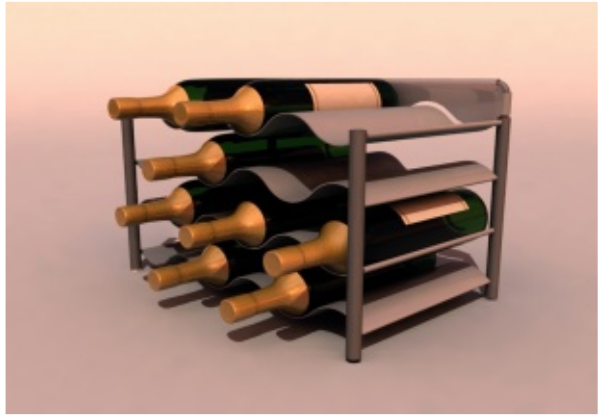
The "White Space" method above had a completely uniform white light, which means the sun lamp used for the additional hard shadow could be placed anywhere. However, most HDR probes have one or more concentrated light sources. First render the scene with only the GI lighting and no lamps. See which way the soft shadows are cast, and then position the sun lamp accordingly. You might need several test renders to get the hard shadow pointing the right way. I usually use very subtle, weak light(s) with HDR maps, the energy should be around 0.2-0.4, not more.



Render settings

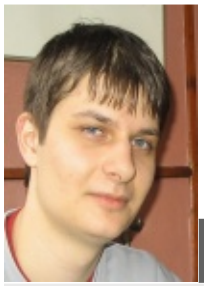
They should be similar to the first method, except for the GI Power setting. This can vary a lot with the given HDR map you are using, depending on how bright it is. Do some low res quick renders to find the right brightness. Some darker maps need 1-1.5 at the most, while the brighter ones might only need a GI Power of 0.2. And then comes the final render! Here is my test scene with a sunset HDR probe made with Terragen. This same quality without GI would need a few dozen lamps, all set up properly, so we clearly see the advantages.

Not only do HDR maps provide the base lighting of a scene, but they also provide a realistic environment the objects can reflect. They can also be the background, however most HDR probes are too low-res to provide a nice background image. Instead, it is preferable to use a photo of the original location for a background. This test image shows one of my own probes in action, along with a photo background taken at the same location. Note that instead of a grey plane which screams CGI, I used a stone pedestal, making it blend in better.



This was just one small portion of what you can do with global illumination, you can also use it for indoor scenes. Try placing emitting planes or area lamps in your scene. For caustics, add "photon lamps", which are available in the Lamp settings in Blender. ■

Experiment. Have fun!



Zsolt Stephan

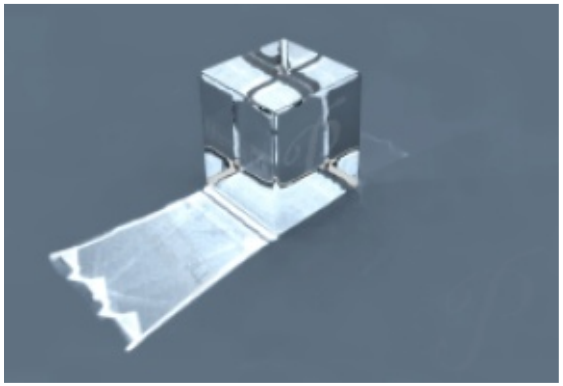
I live in Hungary, and am currently studying industrial design engineering at the Budapest University of Technology and Economics.

Some of my hobbies are: 3D graphics of course :), concept design/product design, plus watching good movies, going out with friends, hanging out on elysiun and other forums, drawing.

http://yafray.org

Using Yafrays caustics system from within Blender

by Daniel LaBarge



In this tutorial we are going to experiment with Yafray and Blender and learn how they interact together. We will create a simple caustic effect using default settings. Further experimentation can be used to create breath taking imagery!

Getting Started

First, we will need to have the latest versions of Blender [Blender 2.41] and Yafray [Yafray 0.7] both installed on your computer. You can get these from www.blender.org and www.yafray.org respectively. Once installed, you will be ready to begin rendering your First Caustics scene!

Using Caustics

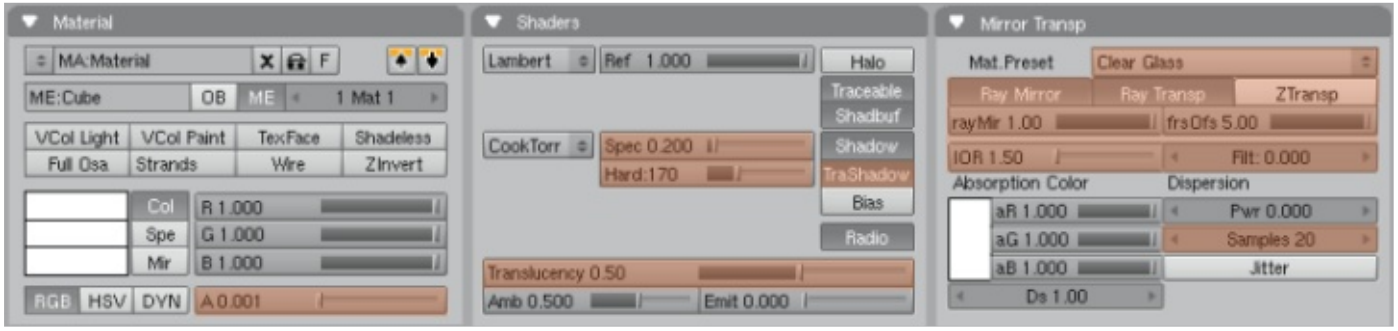
The setup of the scene is very simple. We will be using the default scene with some lamp placement changes.

Step1. Launch Blender.
Step2. Go to the Render Buttons [F10] and switch the Renderer from Blender Internal to the Yafray Renderer.

This will create some new panels in the Render Buttons, Edit Buttons, Material Buttons, and the Lamp Buttons. We will get to these new panels shortly.

Step3. In the 3D Window Select [RMB] the default cube. Go to the Material Buttons [F8]. The Raytracing Panel introduces the available Yafray features. We will be making a clear glass using default settings for glass in material panel. Select 'Clear Glass' from the drop down menu. Set the remainder of your material settings to Fig. 1. You might want to bevel the edges of the mesh. For this object I used Subdivide Multi 3X and then added a 0.100 Bevel 2X and then applied Set Smooth. A basic bevel would work just as well. Watch the image at bottom.

Step4. In the 3D Window switch to Top View [Num7] and Add [SpaceBar] a Plane [Add>>Mesh>>Plane] to the scene, this is going to be the floor in the scene.. Exit Edit Mode [Tab]. You will need to Move [G] it down along the Z-axis [Z] one grid unit [Hold Ctrl] so that it is flush with bottom of the Cube. Open the Transform Properties Panel [Nkey] and scale the Plane to 50 units or simply press [S] and scale it to spread up to the camera view. Go to the Material Buttons [F8] and Add a New Material.



Make sure to have the TraceShadow feature enabled in this material. You can even apply a texture if you like. I used a light blue solid color for this material.

Step5. In the 3D window switch to Top View [Num7] and select [RMB] the default lamp and delete [Del] it. Now place your 3D Cursor at approximately 0,4,2 XYZ Global coordinates. Press [SpaceBar] to add an Area Lamp Add>>Lamp>>Area. The lamp should be automatically selected, but if not select it and then bring up the Transform Properties Panel [N]. Set the Location XYZ coordinates to 0,4,2 respectfully in the transform panel. Set the Rotation XYZ coordinates to 65,0,180 respectfully. In the Material/Lamp Buttons set the power to 0.250. Give it the shape of a Square 3.0. Set the 'Shadow Samples' at 10 or greater.

Step6. Duplicate [Shift+D] the Area Light then Cancel [RMB] the Auto-Grab function. This will place the duplicated area light at the exact location of the original area light. In the Material/Lamp buttons change the lamp type to Photon Lamp. In Blender only way to generates caustics in the scene is using this lamp. Change the Photon settings as in Fig. 2.

Step7. Now switch over to Render Buttons [F10] and change these settings as in Fig. 3.

Step8. Render [F12] the scene! Your results should be similar to first image in previous page. If your render doesn't match it try to match closely the settings as given earlier in the figures. You can also look at the example caustics.blend file if you can't figure it out!

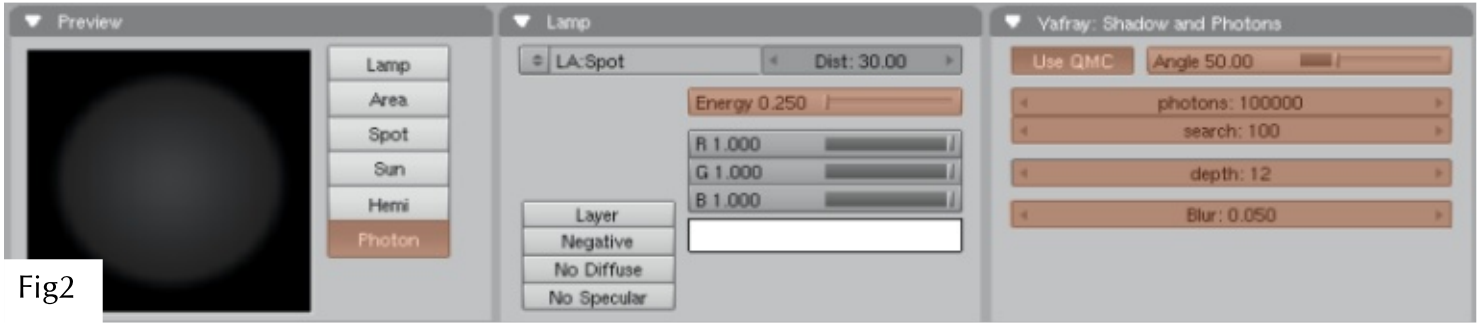


Fig2

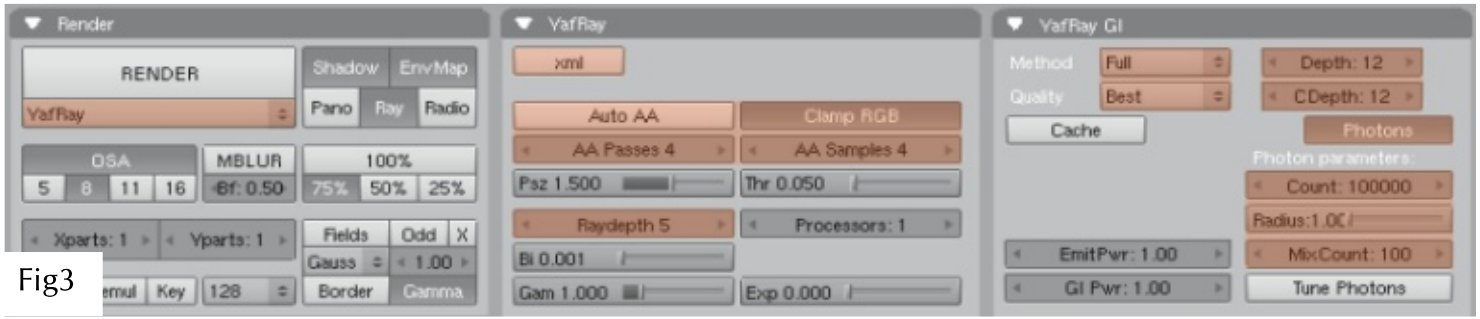


Fig3

What Just Happened?

Now that we have generated a good example of Caustics, now we are going to look at how the Caustics worked in this scene and what else options are available in the blender interface. There are three panels in blender interface that contain settings for Caustics:

The first one is the Yafray Panel itself. This panel allows you to control the Yafray Renderer settings. Normally it's good enough to just disable XML so that you can see the interactive rendering. Also you can leave Auto-AA enabled to let Yafray automatically control the anti aliasing option. For detailed control of Caustics you will have to switch to manual anti aliasing setting though.

The second panel is the Yafray GI Panel. GI stands for Global Illumination. Global Illumination is the effect produced by the light being emitted from a Global light source. For this experiment we used the Full Type GI and set the quality to High for better results. Other settings are Depth and CDepth. Depth is the distance that Yafray calculates a single ray during raytracing. The smaller the value the quicker but less accurate the raytracing will be. CDepth is similar but it is for caustics photons in the scene. Again the smaller the value the quicker but less accurate the caustics in the scene will be. We also enabled Photons in this panel. This will enable GI Photons which are also referred to as Helper Photons. They then are emitted from all of the lights in the scene. The more photons (we used 100,000) better the results, but this will also exponentially increase the rendering time.

The last panel is the Lamp Panel[F5]. This is located in the Material/Lamp buttons. You will need to select the photon lamp in the scene to see the Yafray Shadow and Photon tab. By enabling Photon Lamp we enable the use of photons for the caustics from a direct source. Here the Power setting is like any other light and tells how "bright" the caustics will be. This photon lamp allows us to direct the direction of caustic and optimize the caustics results nicely. We used about 100,000 photons here. For some complex scene it can go all the way up to a million! It just depends on how much computation you want to do! You can experiment with the other settings as they usually require some testing but default settings will usually achieve good results regardless.

Extending Caustics

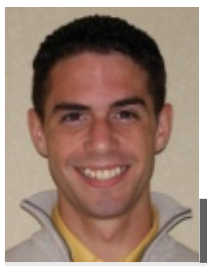
Caustics are not restricted to Transparent materials either! You can use caustics for metallic materials too, just remember that two metallic materials will also reflect the photons so they generally will not show up some times.

To finish up this tutorial I'll give you some suggestions to play around with:

Try adding a texture or a color to the Cube and giving it some Nor or Displacement.

Replace the Cube with a Sphere or another mesh. Use several Photon Lamps. Have your super cluster handy! Use an HDR Global Texture.

Email me your experiments! I'd love to take a look at them! ■



Daniel LaBarge

I am a 17 year old graphics artist and programmer from San Antonio, Texas.

A freelance studio artist at ID Studios [www.intellidesign.org]
and a web programmer at MonsterWeb www.monsterweb.net

DPI demystified

You did it! You have truly created a masterpiece. But when you printed it out, it looked all pixilated and blocky. Dang, you must have a bad setting somewhere. You go back, double check all your settings, hit render again and wait. It still looks the same when you print it. You start glaring silent threats at your printer. Well, before you decide to make good on those threats, you might want to try one more thing (your printer will thank you for it).

You may have overlooked the whole DPI issue. While it's true that Blender has some nice preset render sizes, if you don't understand how that relates to printing, you will be forever glaring at your printer.

First let's look at what DPI actually is.
DPI is a printing term that describes the number of dots/pixels that are defined in the boundary of a square inch that are used to create an image. The more correct term is pixels per inch, however dots per inch is often used instead (The image can be a font or graphic). In general, the more dots, the better and sharper the image. DPI is printer resolution. DPI is not image resolution although frequently used that way.
Ok, now that we know what DPI is, let's look at how it applies to Blender. Blender renders images out at 72dpi, which looks okay on your screen, but not printed out. In order for it to look good for print, you have to render your image bigger and then resize it smaller in your favorite graphic program. Essentially what you will be doing is swapping physical dimensions for resolution. Take a look at the following chart (fig. 1 below). It shows some common render sizes in pixels and the corresponding physical sizes at 72dpi (which Blender uses) and a good print resolution of 300dpi. You can see that the physical sizes become smaller as you increase the resolution.

Pixel Dimensions	Image Resolution	Printed size
640 x 480	72 dpi	8.89" x 6.67"
	300 dpi	2.13" x 1.60"
800 x 600	72 dpi	11.11" x 8.33"
	300 dpi	2.67" x 2.00"
1024 x 768	72 dpi	14.22" x 10.67"
	300 dpi	3.41" x 2.56"
1280 x 960	72 dpi	17.78" x 13.33"
	300 dpi	4.27" x 3.20"
1600 x 1200	72 dpi	22.22" x 16.67"
	300 dpi	5.33" x 4.00"
2400 x 1600	72 dpi	33.33" x 22.22"
	300 dpi	8.00" x 5.33"

Fig1

This may seem overly complicated, but really it is not. There is a very simple formula you can use to get the proper settings. Basically you multiply the desired resolution by the desired physical size to get the pixel settings/size you need for Blender (fig. 2).

Once you have rendered your image using those settings, you need to go into your favorite image program and resize your image. Most image programs work similarly, so figuring it out in your program shouldn't be difficult. I will be using Photoshop to explain this part, because that is what I have.

Desired Printed Size	(x)	Desired Resolution	(=)	Needed Pixel Size in Blender
8 x 10 inches		150		1200 x 1500
8 x 10 inches		200		1600 x 2000
8 x 10 inches		300		2400 x 3000

Fig2

Step 1: Open your image

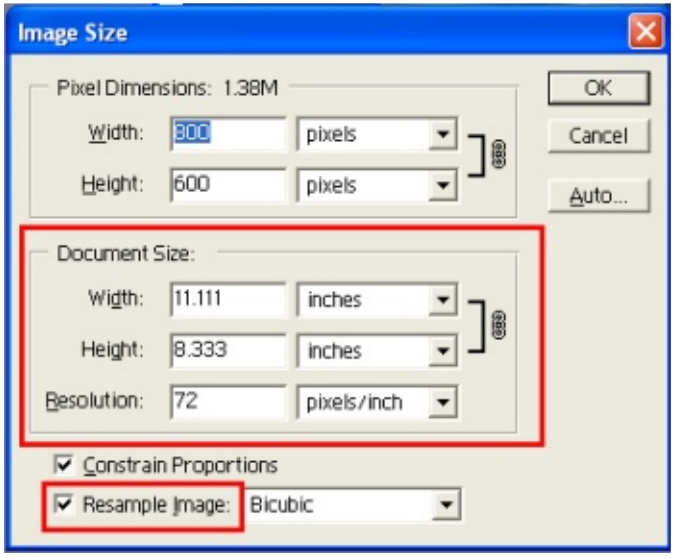
Step 2: Go to Image>Image Size, a dialog box like fig. 3 should open.

Step 3: At the bottom of the box, uncheck “Resample Image”

Step 4: Next locate the “Resolution Box”, enter your new resolution. The Width and Height boxes should automatically update to reflect the new size at that resolution.

Step 5: Hit the “OK” button and save, you are all done and now your image will print like you expected it to.

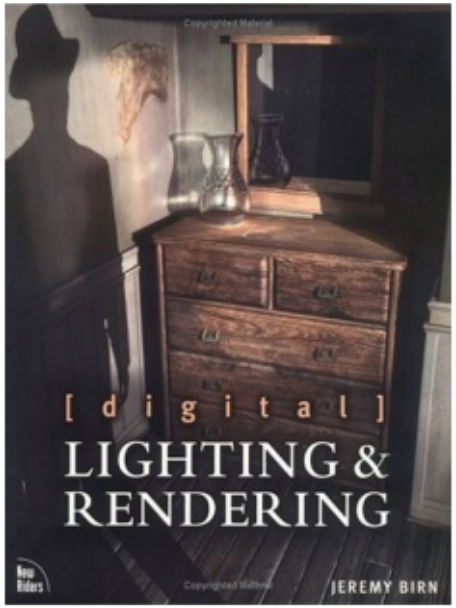
Okay, so now you have a better understanding of dpi and how to get the proper dpi for printing, but what dpi resolution is best for what you need. As a general rule, you should choose a resolution between 150-300 dpi. This resolution is good for most print projects. If you are having something professionally printed, it is always a good idea to check with your printer and find out what resolution their equipment runs at.



Sandra Gilbert

Hi, my name is Sandra Gilbert, (aka dreamsgate). I have been using blender for a little over five years. I currently live in Nampa, Idaho, USA and work as a graphic designer for a small print shop.

I’m married and have 2 children, which leaves not near enough time for feeding my growing blender obsession. Yet somehow I always manage to find the time to explore new features, keep up on the latest news and start new blender projects. Some of them I actually even manage to finish.



Digital lighting & rendering

Book Review

Digital Lighting and Rendering

This issue we are reviewing “[digital] Lighting and Rendering” by Jeremy Birn.

By now we have all figured out, or soon will, that lighting is a very large part of any successful project. You can have the best models, materials and composition, but without proper lighting your image will fall flat.

Enter “[digital] Lighting and Rendering”. Jeremy Birn takes you step by step, with logical and clearly explained examples, through the often complex and confusing area of lighting. He starts you off with how to set up a good light workflow and then explains each type of light you should have available.

He continues with in-depth discussions on what lighting is and how light and shadow behave in the real world, then progresses through color theory, quality of light and shadow and exposure.

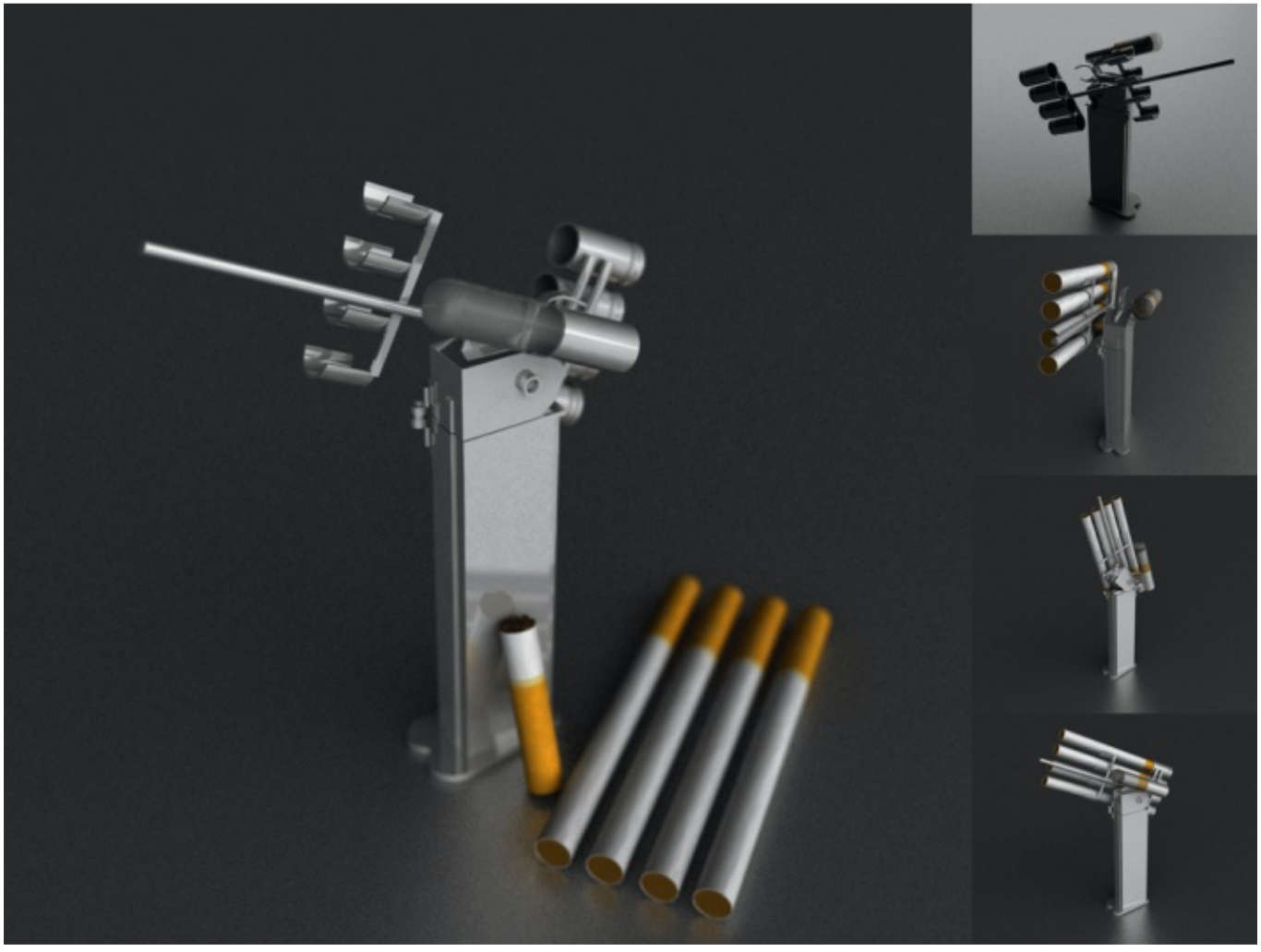
No book on lighting would be any good without a thorough discussion of light setups. And this one is an excellent resource for learning about the various light setups and their uses.
He goes on to explain how materials fit into the rendering pipeline. Starting with shading surfaces, he works his way through detailed explanations of how different material setups affect lighting and rendering. In this same chapter he also explains the uses and advantages of raytracing and global illumination.

The final chapter covers compositing and rendering in layers, showing the use of different render passes to get just the effect you are looking for.

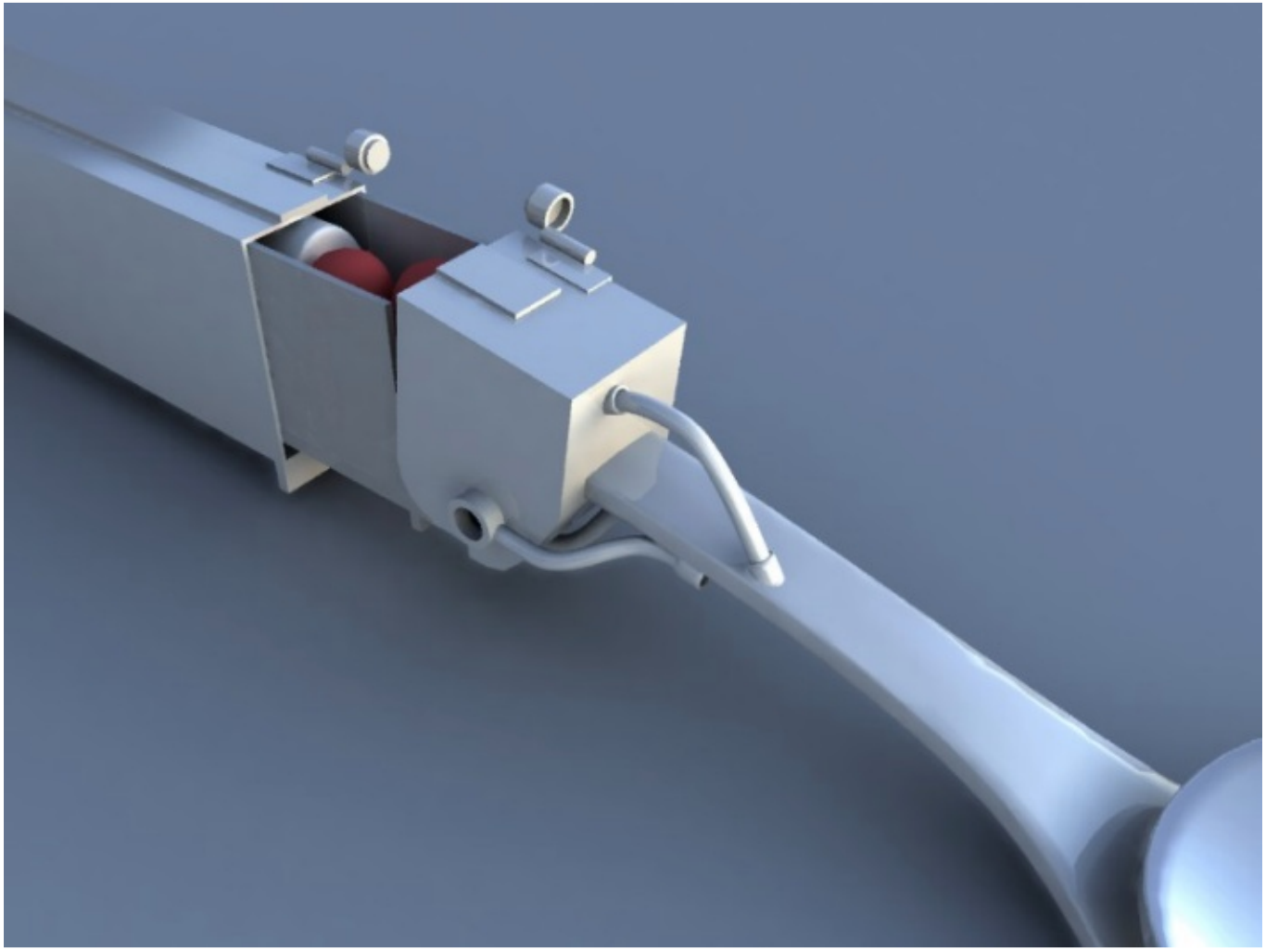
This book is a definite “must read” for beginners and yet even an experienced CG artist will find it worth a look. ■

--blenderart

The wild bean [Yafray]
by - Claas Eicke Kuhnen



Composition [Yafray]
by - Claas Eicke Kuhnen



Pez [Yafray]
by - Claas Eicke Kuhnen

A temple [Yafray]
by - Karan Sah

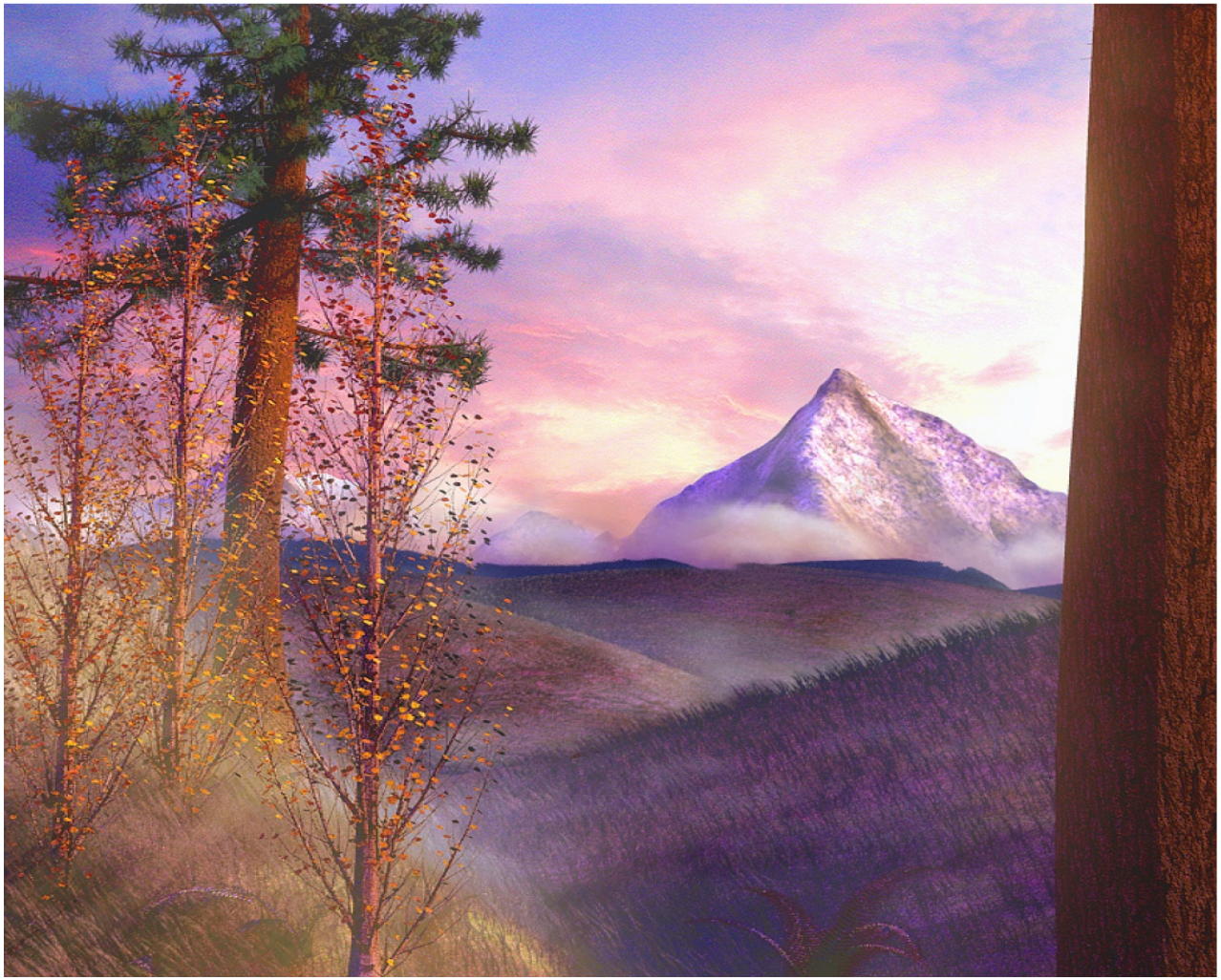
Trishul (trident) [Yafray]
by - Karan Sah

Airsurfboarding girl
by - Zoltan Miklosi

The Assasin
by - Zoltan Miklosi

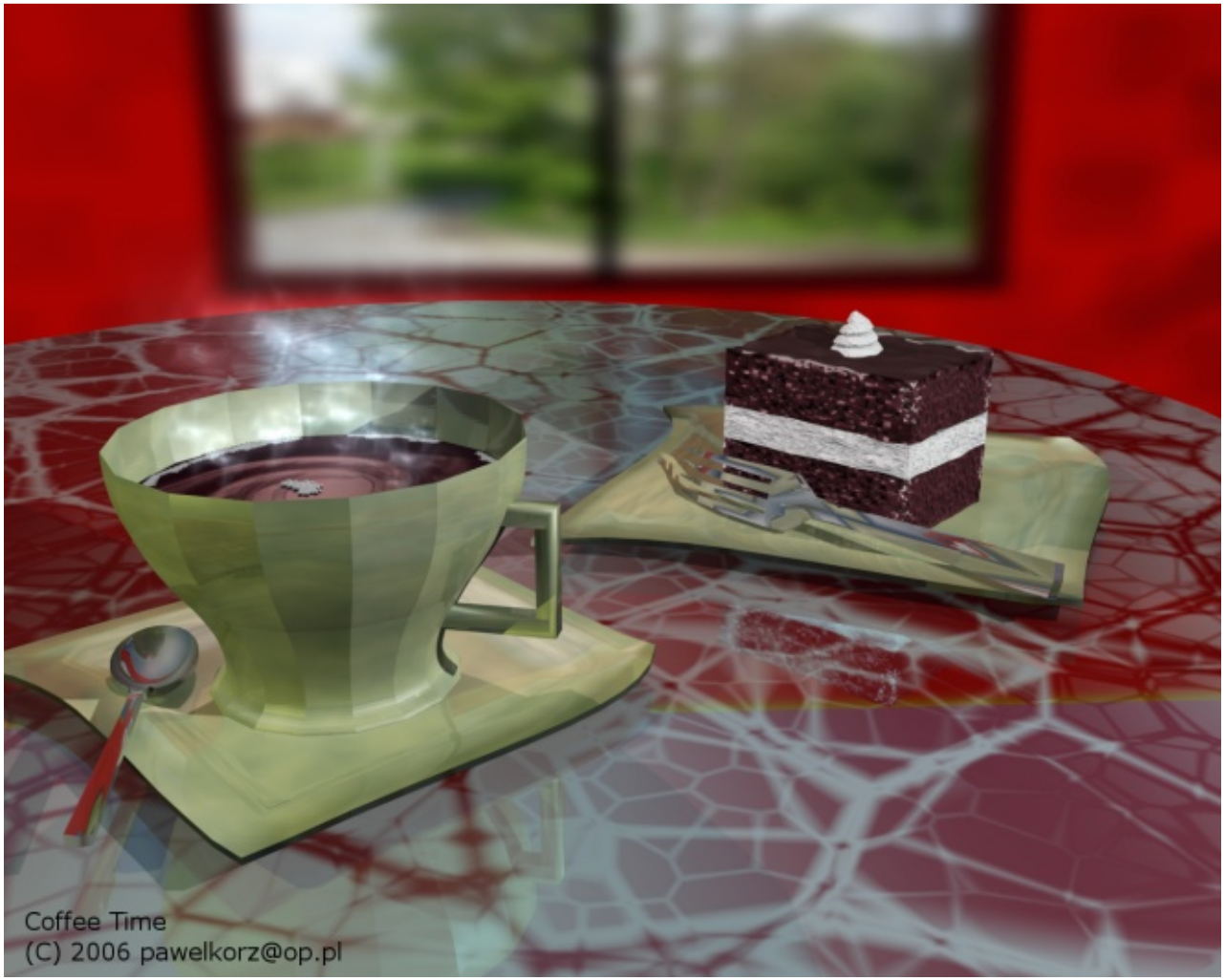
Nancy on the street
by - Zoltan Miklosi

Maluda The Great Warrior
by - Zoltan Miklosi



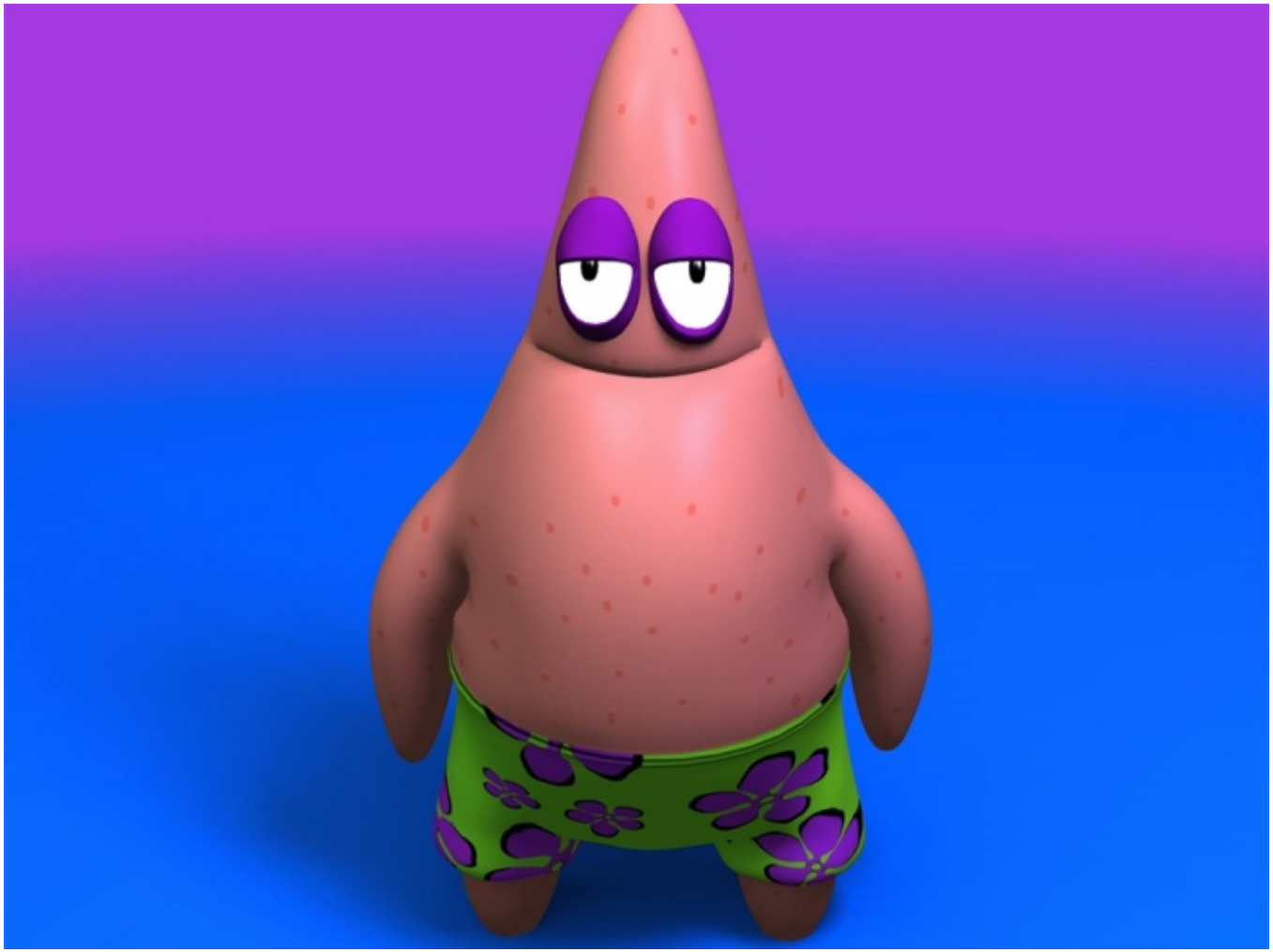
Autumn
by - Jason Saundersi

Abandoned Subway
by - Stephen Davis



Coffee Time
by - Paweł Korzeniak

Alien
by - Derek Marsh



Patrick star
by - John Starr aka itraenk

We have been overwhelmed by the constant apprasial and feedback we got from the blender commuity.

Hello! dear blender heads.

We have been bringing forth a bundle of good learning material for you for the past six months. In every issue, we take on a particular aspect of CG related to Blender and present you with a pack of useful learning material. All this is managed with the help of the experienced blender heads from within the blender community. They help spread the knowledge by sharing content from their respective field of expertise.

Back when we just started the magazine, it was our aim to provide a good resource to cover the different aspects of CG with the main focus being on the ease of learning for the new users. We have been overwhelmed by the constant appraisal and feedback we got from the blender community. That shows that we have at least been successful in that area. I take this opportunity on behalf of our small team at blenderart to thank you all for your comments and feedback and at times, patience. And also to the folks who have provided much needed material for the magazine.

This time we got our website hosted thanks to ibiblio.org which graciously provided us with the web space. Nam Pham did a great job with the new website design and he has also provided the long wanted solution for subscribing to our mailing list. Now all those who will subscribe to it will get notified as soon as the new issue is launched.

With this issue out we are also starting to restructure both the magazine design and the contents so that we can continue to provide you all with even better learning experience for the time to come.

So until then do keep sending us your valuable feedback and comments.

Happy blending!
Gaurav
gaurav@blenderart.org

ISSUE 4

Available in May 2006
Theme: Character Modeling.

Modeling techniques.
Tips and tricks.
